Assignment 1: Foundations Reflection

Introduction

In this assignment, I have compared two different types of machine learning models: a **discriminative model** (Logistic Regression) and a **generative model** (Generative Adversarial Network - GAN). The goal is to understand how they differ in training, behavior, and applications. Logistic regression focuses on classification, while GAN focuses on generating new data. By applying both models to the MNIST dataset, we can observe how simple and stable discriminative models compare with complex and unstable generative models.

Methods

Dataset: I have used the **MNIST dataset** from PyTorch, which contains handwritten digit images (0–9), **size** 28 × 28 pixels, **Grayscale** and labeled with its digit class.

For Logistic Regression: Images were flattened into vectors of size 784. Pixel values were normalized to [0,1].

For GAN: Images were scaled to the range [-1,1] to match the generator's output.

Models:

Logistic Regression (Discriminative Model):

- A linear classifier.
- Predicts the probability of each digit class.
- Trained using labeled data.
- Evaluated using accuracy and confusion matrix.

GAN (Generative Model):

- It consists of two networks: **Generator** (creates fake images) and **Discriminator** (decides if images are real or fake).
- Trained using PyTorch.
- Evaluated visually using generated samples and loss curves.

Training Strategy

Logistic Regression:

For logistic regression, the training process was simple and stable. The MNIST images were first flattened into 784-dimensional vectors and normalized to the range [0,1]. The dataset was then split into training and testing sets. The model was trained using labeled images so it could learn how to correctly identify each digit. The Performance was evaluated using standard metrics such as accuracy and the confusion matrix. The training was done quickly and produced consistent results, making logistic regression easy to implement and interpret.

GAN:

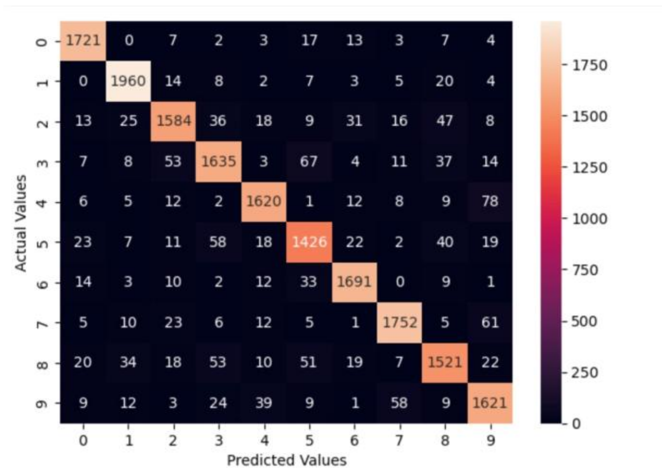
For the GAN, training was carried out using PyTorch and followed an adversarial learning approach. The images were scaled to the range [-1,1] to match the output of the generator. In each training step, the discriminator was first trained to distinguish real images from fake ones produced by the generator. The generator was then trained to produce images that would fool the discriminator into thinking they were real. This alternating training process continued for several epochs. The performance of the GAN was evaluated using the generator and discriminator loss curves and visually inspecting the generated images, since there is no direct accuracy measure for generative models.

Results

Logistic Regression:

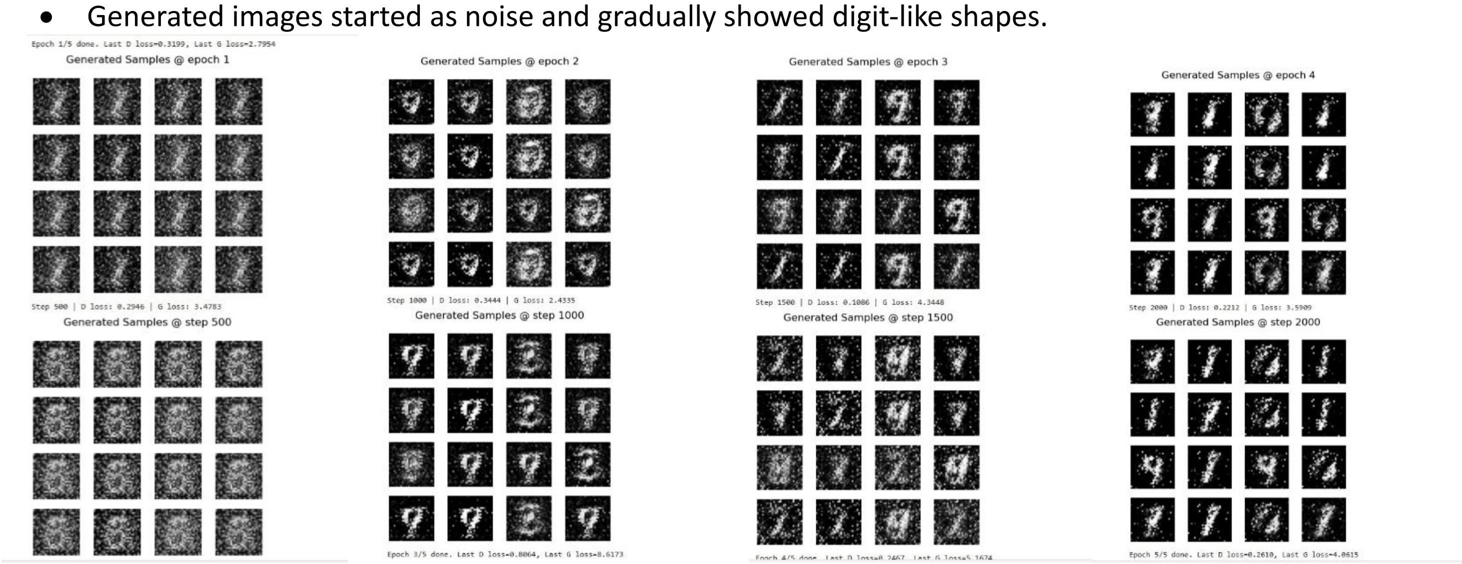
- Achieved **92% accuracy.**
- Most predictions lie on the diagonal of the confusion matrix.
- Logistic regression performs well and is reliable for digit classification.

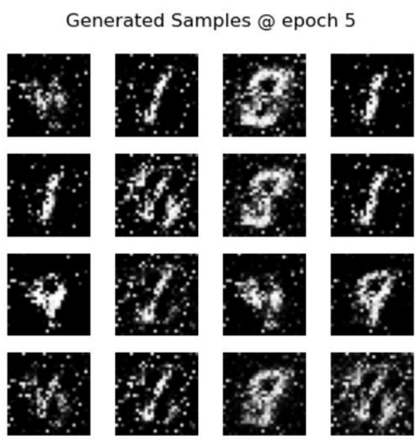
	precision	recall	f1-score	support
0	0.95	0.97	0.96	1777
1	0.95	0.97	0.96	2023
2	0.91	0.89	0.90	1787
3	0.90	0.89	0.89	1839
4	0.93	0.92	0.93	1753
5	0.88	0.88	0.88	1626
6	0.94	0.95	0.95	1775
7	0.94	0.93	0.94	1880
8	0.89	0.87	0.88	1755
9	0.88	0.91	0.90	1785
accuracy			0.92	18000
macro avg	0.92	0.92	0.92	18000
weighted avg	0.92	0.92	0.92	18000

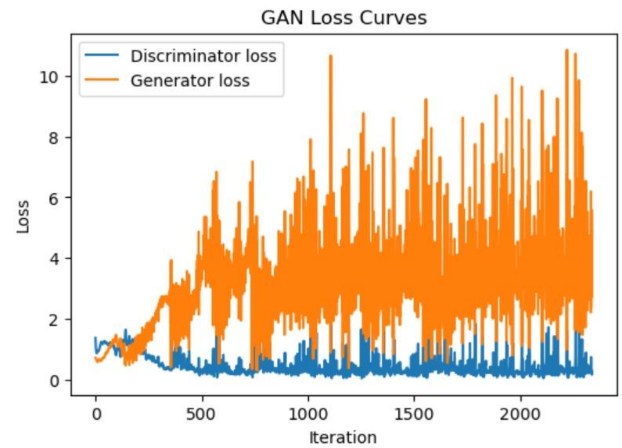


GAN:

- Loss curves oscillated and did not converge smoothly.
- Discriminators learned faster than the generator.
- Generated images started as noise and gradually showed digit-like shapes.







Reflection and Analysis

Training and comparing logistic regression and the GAN showed the difference between discriminative and generative models. Logistic regression was easy to train, stable, and predictable, with performance measured using a simple accuracy score and confusion matrix. It was easy to see when the model was improving and where it made mistakes. The GAN was much more difficult to train because it involved two competing networks. Its training was unstable and sensitive to hyperparameters and did not converge smoothly. There was no single numerical metric, such as accuracy, to evaluate its performance, so the model's quality had to be determined primarily by visually inspecting the generated images. This made the GAN more experimental and difficult to debug. Overall, this comparison revealed that classification tasks are simpler and more reliable to implement, whereas generative tasks are more complex but powerful, allowing models to generate entirely new data rather than simply recognizing existing patterns.

Conclusion

Discriminative models, such as logistic regression, are simple, stable, and reliable for prediction tasks. Generative models, such as GANs, are powerful but difficult to train, and are better suited to creative and data generation tasks.

Logistic regression is used for:

- Digit recognition
- Spam detection
- Medical diagnosis
- Fraud detection

GAN is used for:

- Image generation
- Data augmentation
- Art and creativity
- Synthetic data creation
- Simulation and modeling